

# Package ‘ngsReports’

October 27, 2020

**Version** 1.6.0

**Date** 2020/10/02

**Title** Load FastqQC reports and other NGS related files

**Description** This package provides methods and object classes for parsing FastQC reports and output summaries from other NGS tools into R, as well as visualising the data loaded from these files.

**URL** <https://github.com/steveped/ngsReports>

**BugReports** <https://github.com/steveped/ngsReports/issues>

**License** file LICENSE

**Encoding** UTF-8

**Depends** R (>= 4.0.0),  
BiocGenerics,  
ggplot2,  
tibble (>= 1.3.1)

**Imports** Biostrings,  
checkmate,  
dplyr (>= 1.0.0),  
DT,  
FactoMineR,  
forcats,  
ggdendro,  
grDevices,  
grid,  
lifecycle,  
lubridate,  
methods,  
pander,  
parallel,  
plotly,  
readr,  
reshape2,  
rmarkdown,  
Rsamtools,  
scales,  
ShortRead,  
stats,  
stringr,

```
tidyr,
tidyselect (>= 0.2.3),
utils,
viridisLite,
zoo
```

**LazyData** true

**RoxygenNote** 7.1.1

**Collate** 'AllGenerics.R'  
 'validationFunctions.R'  
 'FastqcData.R'  
 'FastqcDataList.R'  
 'FastqcFile.R'  
 'PwfCols.R'  
 'S4coercion.R'  
 'TheoreticalGC.R'  
 'aaa.R'  
 'data.R'  
 'errMsg.R'  
 'estGcDistn.R'  
 'extract.R'  
 'fqName.R'  
 'fqcVersion.R'  
 'getColours.R'  
 'getGC.R'  
 'getModule.R'  
 'getSummary.R'  
 'helpers.R'  
 'importNgsLogs.R'  
 'importSJ.R'  
 'isCompressed.R'  
 'maxAdapterContent.R'  
 'ngsReports-package.R'  
 'overRep2Fasta.R'  
 'path.R'  
 'plotAdapterContent.R'  
 'plotAlignmentSummary.R'  
 'plotAssemblyStats.R'  
 'plotBaseQuals.R'  
 'plotDupLevels.R'  
 'plotFastqcPCA.R'  
 'plotGcContent.R'  
 'plotKmers.R'  
 'plotNContent.R'  
 'plotOverrep.R'  
 'plotReadTotals.R'  
 'plotSeqContent.R'  
 'plotSeqLengthDistn.R'  
 'plotSeqQuals.R'  
 'plotSummary.R'  
 'pwf.R'  
 'readTotals.R'  
 'runFastQC.R'

'writeHtmlReport.R'

**VignetteBuilder** knitr

**Suggests** BiocStyle,

Cairo,

knitr,

testthat,

truncnorm

**biocViews** QualityControl, ReportWriting

**RdMacros** lifecycle

**git\_url** <https://git.bioconductor.org/packages/ngsReports>

**git\_branch** RELEASE\_3\_12

**git\_last\_commit** bdad1ba

**git\_last\_commit\_date** 2020-10-27

**Date/Publication** 2020-10-27

## R topics documented:

estGcDistn . . . . .	4
FastqcData-class . . . . .	5
FastqcDataList-class . . . . .	6
fqcVersion, FastqcData-method . . . . .	7
fqName . . . . .	8
gcAvail . . . . .	8
gcTheoretical . . . . .	9
getColours . . . . .	10
getGC . . . . .	11
getModule, FastqcData-method . . . . .	12
getSummary, FastqcFile-method . . . . .	13
importNgsLogs . . . . .	14
importSJ . . . . .	15
isCompressed . . . . .	16
maxAdapterContent . . . . .	16
mData . . . . .	17
overRep2Fasta . . . . .	18
path . . . . .	19
plotAdapterContent . . . . .	20
plotAlignmentSummary . . . . .	22
plotAssemblyStats . . . . .	23
plotBaseQuals . . . . .	24
plotDupLevels . . . . .	26
plotFastqcPCA . . . . .	28
plotGcContent . . . . .	30
plotKmers . . . . .	33
plotNContent . . . . .	35
plotOverrep . . . . .	36
plotReadTotals . . . . .	38
plotSeqContent . . . . .	40
plotSeqLengthDistn . . . . .	42
plotSeqQuals . . . . .	43

plotSummary . . . . .	46
pwf . . . . .	48
PwfCols-class . . . . .	48
readTotals . . . . .	49
runFastQC . . . . .	50
TheoreticalGC-class . . . . .	53
writeHtmlReport . . . . .	53
[ . . . . .	55

<b>Index</b>	<b>56</b>
--------------	-----------

---

estGcDistn	<i>Estimate a GC Content Distribution From Sequences</i>
------------	----------------------------------------------------------

---

## Description

Generate a GC content distribution from sequences for a given read length and fragment length

## Usage

```
estGcDistn(x, n = 1e+06, rl = 100, fl = 200, fragSd = 30, bins = 101, ...)
```

```
## S4 method for signature 'ANY'
```

```
estGcDistn(x, n = 1e+06, rl = 100, fl = 200, fragSd = 30, bins = 101, ...)
```

```
## S4 method for signature 'character'
```

```
estGcDistn(x, n = 1e+06, rl = 100, fl = 200, fragSd = 30, bins = 101, ...)
```

```
## S4 method for signature 'DNAStringSet'
```

```
estGcDistn(x, n = 1e+06, rl = 100, fl = 200, fragSd = 30, bins = 101, ...)
```

## Arguments

x	DNAStringSet or path to a fasta file
n	The number of reads to sample
rl	Read Lengths to sample
fl	The mean of the fragment lengths sequenced
fragSd	The standard deviation of the fragment lengths being sequenced
bins	The number of bins to estimate
...	Not used

## Details

The function takes the supplied object and returns the theoretical GC content distribution. Using a fixed read length essentially leads to a discrete distribution so the bins argument is used to define the number of bins returned. This defaults to 101 for 0 to 100

The returned values are obtained by interpolating the values obtained during sampling. This avoids returned distributions with gaps and jumps as would be obtained setting readLengths at values not in multiples of 100.

Based heavily on <https://github.com/mikelove/fastqcTheoreticalGC>

**Value**

A tibble with two columns: GC\_Content and Freq denoting the proportion of GC and frequency of occurrence respectively

**Examples**

```
faDir <- system.file("extdata", package = "ngsReports")
faFile <- list.files(faDir, pattern = "fasta", full.names = TRUE)
df <- estGcDistn(faFile, n = 200)
```

---

FastqcData-class	<i>The FastqcData Object Class</i>
------------------	------------------------------------

---

**Description**

The FastqcData Object Class **Stable**

**Usage**

```
FastqcData(x)
```

**Arguments**

x Path to a single zip archive or extracted folder for a individual FastQC report.

**Details**

This object class is the main object required for generating plots and tables. Instantiation will first test for a compressed file (or extracted directory) with the correct data structure, and will then parse all the data into R as a FastqcData object. FastQC modules are contained as individual slots, which can be viewed using slotNames.

Individual modules can be returned using the function getModule() and specifying which module is required. See [getModule](#) for more details.

**Value**

An object of class FastqcData

**Slots**

Summary Summary of PASS/WARN/FAIL status for each module

Basic\_Statistics The Basic\_Statistics table from the top of a FastQC html report

Per\_base\_sequence\_quality The underlying data from the Per\_base\_sequence\_quality module

Per\_sequence\_quality\_scores The underlying data from the Per\_sequence\_quality\_scores module

Per\_base\_sequence\_content The underlying data from the Per\_base\_sequence\_content module

Per\_sequence\_GC\_content The underlying data from the Per\_sequence\_GC\_content module

Per\_base\_N\_content The underlying data from the Per\_base\_N\_content module

Sequence\_Length\_Distribution The underlying data from the Sequence\_Length\_Distribution module

Sequence\_Duplication\_Levels The underlying data from the Sequence\_Duplication\_Levels module

Overrepresented\_sequences The underlying data from the Overrepresented\_sequences module

Adapter\_Content The underlying data from the Adapter\_Content module

Kmer\_Content The underlying data from the Kmer\_Content module

Total\_Deduplicated\_Percentage Estimate taken from the plot data for Sequence\_Duplication\_Levels. Only included in later versions of FastQC

version The version of FastQC used for generation of the report (if available)

path Path to the FastQC report#'

## Examples

```
# Get the files included with the package
packageDir <- system.file("extdata", package = "ngsReports")
fl <- list.files(packageDir, pattern = "fastqc.zip", full.names = TRUE)[1]

# Load the FASTQC data as a FastqcData object
fd <- FastqcData(fl)
fd
```

---

FastqcDataList-class    *The FastqcDataList Object Class*

---

## Description

The FastqcDataList Object Class **Stable**

## Usage

```
FastqcDataList(x)
```

## Arguments

x                      Character vector of file paths specifying paths to FastQC reports

## Value

An object of class FastqcDataList

## Slots

... this can either be a single character vector of paths to FASTQC files, or several instances of .FastqcFile objects

## Examples

```
# Get the files included with the package
packageDir <- system.file("extdata", package = "ngsReports")
fl <- list.files(packageDir, pattern = "fastqc.zip", full.names = TRUE)

# Load the FASTQC data as a FastqcDataList object
fdl <- FastqcDataList(fl)
fdl
```

---

fqcVersion, FastqcData-method  
*Get the FASTQC version*

---

## Description

Get the FASTQC version used to generate the initial files

## Usage

```
## S4 method for signature 'FastqcData'
fqcVersion(object)

## S4 method for signature 'FastqcDataList'
fqcVersion(object)

## S4 method for signature 'ANY'
fqcVersion(object)
```

## Arguments

object                    An object of class FastqcData or FastqcDataList

## Value

A character vector (FastqcData), or tibble (FastqcDataList)

## Examples

```
# Get the files included with the package
packageDir <- system.file("extdata", package = "ngsReports")
fl <- list.files(packageDir, pattern = "fastqc.zip", full.names = TRUE)

# Load the FASTQC data as a FastqcDataList object
fdl <- FastqcDataList(fl)

# Get the FASTQC version
fqcVersion(fdl)
```

---

fqName	<i>Return the Underlying Fastq File Names from FastqcData* Objects</i>
--------	------------------------------------------------------------------------

---

### Description

Return the Underlying Fastq File Names from FastqcData\* Objects

### Usage

```
fqName(object)

## S4 method for signature 'ANY'
fqName(object)

## S4 method for signature 'FastqcData'
fqName(object)

## S4 method for signature 'FastqcDataList'
fqName(object)
```

### Arguments

object            An object of class FastqcData or FastqcDataList

### Value

Returns the names of the Fastq files the FastQC report was generated from, without any preceding directories.

### Examples

```
# Get the files included with the package
packageDir <- system.file("extdata", package = "ngsReports")
fl <- list.files(packageDir, pattern = "fastqc.zip", full.names = TRUE)

# Load the FASTQC data as a FastqcDataList object
fdl <- FastqcDataList(fl)
fqName(fdl)
```

---

gcAvail	<i>List Genomes or Transcriptomes with Theoretical GC Content</i>
---------	-------------------------------------------------------------------

---

### Description

List available genomes or transcriptomes in a TheoreticalGC object



**Usage**

```
gcAvail(object, type)

## S4 method for signature 'TheoreticalGC'
gcAvail(object, type)
```

**Arguments**

object	An object of class TheoreticalGC
type	character indicating either Genome or Transcriptome

**Details**

An object of class TheoreticalGC can hold the theoretical GC content for one or more species, for either the genome or transcriptome. This function checks which species are available in the given object, for either the genome or transcriptome, as supplied to the parameter type.

**Value**

A tibble object

**Examples**

```
gcAvail(gcTheoretical, "Genome")
```

---

gcTheoretical	<i>Theoretical GC content</i>
---------------	-------------------------------

---

**Description**

This object contains the theoretical GC content for each provided species, for both the genome and transcriptome, where available.

**Usage**

```
gcTheoretical
```

**Format**

An object of class TheoreticalGC of length 1.

**Details**

The object is defined with the S4 class TheoreticalGC. Species for which information is available can be found using the command `gcAvail(gcTheoretical)` and selecting the appropriate type.

Metadata is accessible using `mData(gcTheoretical)`.

All GC content was calculated using code from <https://github.com/mikelove/fastqcTheoreticalGC> using BSgenome packages. This provides a default set of GC content data for common organisms generated using 100bp reads/fragments and 1e6 reads.

**See Also**

gcAvail

**Examples**

```
## Check which genomes are included
gcAvail(gcTheoretical, "Genome")

## Check which transcriptomes are included
gcAvail(gcTheoretical, "Transcriptome")
```

getColours

*Work with objects of class PwfCols***Description**

Get and modify colours from objects of class PwfCols

**Usage**

```
## S4 method for signature 'PwfCols'
getColours(object)

## S4 method for signature 'PwfCols'
setColours(object, PASS, WARN, FAIL, MAX)

## S4 method for signature 'PwfCols'
setAlpha(object, alpha)
```

**Arguments**

object	An object of class PwfCols
PASS	The colour denoting PASS on all plots, in rgb format
WARN	The colour denoting WARN on all plots, in rgb format
FAIL	The colour denoting FAIL on all plots, in rgb format
MAX	The colour denoting the limit of values in rgb format
alpha	Numeric(1). Ranges from 0 to 1 by default, but can also be on the range 0 to 255.

**Details**

Use getColours to obtain the colours in an object of class PwfCols.

These can be modified using the functions setColours and setAlpha

**Value**

getColours will return a character vector of colours corresponding to PASS/WARN/FAIL

setColours will return an object of class PwfCols

setAlpha will return an object of class PwfCols

**Examples**

```
getColours(pwf)

# How to add transparency
pwf2 <- setAlpha(pwf, 0.1)
getColours(pwf2)
```

---

getGC	<i>Get Theoretical GC content</i>
-------	-----------------------------------

---

**Description**

Get the GC content data from a TheoreticalGC object

**Usage**

```
getGC(object, name, type)

## S4 method for signature 'ANY'
getGC(object, type)

## S4 method for signature 'TheoreticalGC'
getGC(object, name, type)
```

**Arguments**

object	An object of class Theoretical GC
name	The Name of the species in 'Gspecies' format, e.g. Hsapiens
type	The type of GC content. Can only be either "Genome" or "Transcriptome"

**Value**

A tibble object

**Examples**

```
getGC(gcTheoretical, name = "Hsapiens", type = "Genome")
```

---

getModule, FastqcData-method

*Retrieve a given module from a Fastqc\* Object*


---

## Description

Retrieve a specific module from a Fastqc\* object as a data.frame

## Usage

```
## S4 method for signature 'FastqcData'
getModule(object, module)
```

```
## S4 method for signature 'FastqcDataList'
getModule(object, module)
```

```
## S4 method for signature 'ANY'
getModule(object, module)
```

## Arguments

object	Can be a FastqcData, fastqcDataList, or simply a character vector of paths
module	The requested module as contained in a FastQC report. Possible values are Summary, Basic_Statistics, Per_base_sequence_quality, Per_tile_sequence_quality, Per_sequence_quality_scores, Per_base_sequence_content, Per_sequence_GC_content, Per_base_N_content, Sequence_Length_Distribution, Sequence_Duplication_Levels, Overrepresented_sequences, Adapter_Content, Kmer_Content, Total_Deduplicated_Percent. Note that spelling and capitalisation is exactly as contained within a FastQC report, with the exception that spaces have been converted to underscores. Partial matching is implemented for this argument.

## Details

This function will return a given module from a Fastqc\* object as a data.frame. Note that each module will be it's own unique structure, although all will return a data.frame

## Value

A single tibble containing module-level information from all FastQC reports contained in the Fastqc\* object.

## Examples

```
# Get the files included with the package
packageDir <- system.file("extdata", package = "ngsReports")
fl <- list.files(packageDir, pattern = "fastqc.zip", full.names = TRUE)

# Load the FASTQC data as a FastqcDataList object
fdl <- FastqcDataList(fl)

# Extract the Summary module, which corresponds to the PASS/WARN/FAIL flags
getModule(fdl, "Summary")
```

```
# The Basic_Statistics module corresponds to the table at the top of each
# FastQC report
getModule(fdl, "Basic_Statistics")
```

---

getSummary,.FastqcFile-method

*Get the summary information from Fastqc Files*


---

## Description

Read the information from the summary.txt files in each .FastqcFile

## Usage

```
## S4 method for signature '.FastqcFile'
getSummary(object)

## S4 method for signature 'ANY'
getSummary(object)

## S4 method for signature 'FastqcData'
getSummary(object)

## S4 method for signature 'FastqcDataList'
getSummary(object)
```

## Arguments

object	Can be a FastqcData, FastqcDataList object or a vector of paths to unparsed FastQC reports.
--------	---------------------------------------------------------------------------------------------

## Details

This simply extracts the summary of PASS/WARN/FAIL status for every module as defined by the tool FastQC for each supplied file.

## Value

A tibble containing the PASS/WARN/FAIL status for each module, as defined in a FastQC report.

## Examples

```
# Get the files included with the package
packageDir <- system.file("extdata", package = "ngsReports")
fl <- list.files(packageDir, pattern = "fastqc.zip", full.names = TRUE)

# Load the FASTQC data as a FastqcDataList object
fdl <- FastqcDataList(fl)

# Return a tibble/tibble with the raw information
```

```
getSummary(fdl)
```

---

importNgsLogs

---

*Import Various NGS-related log files*


---

## Description

Imports NGS-related log files such as those generated from stderr **Maturing**

## Usage

```
importNgsLogs(x, type = "auto", which, stripPaths = TRUE)
```

## Arguments

x	character. Vector of filenames. All log files must be of the same type. Duplicate file paths will be silently ignored.
type	character. The type of file being imported. Can be one of bowtie, bowtie2, hisat2, star, flagstat, featureCounts, duplicationMetrics, cutadapt, adapterRemoval, quast or busco. Defaults to type = "auto" which will automatically detect the file type for all implemented types.
which	Which element of the parsed object to return. Ignored in all file types except when type is set to duplicationMetrics, cutadapt or adapterRemoval. See details for possible values
stripPaths	logical(1). Remove paths from the Filename column

## Details

Imports one or more log files as output by tools such as: bowtie, bowtie2, featureCounts, Hisat2, STAR, picard MarkDuplicates, cutadapt, flagstat, Adapter Removal, trimmomatic quast or busco. autoDetect can be used to detect the log type by parsing the file.

The featureCounts log file corresponds to the counts.out.summary, not the main counts.out file.

Whilst most log files return a single tibble, some are more complex with multiple modules.

adapterRemoval can return one of four modules (which = 1:4),. When calling by name, the possible values are sequences, settings, statistics or distribution. Partial matching is implemented.

cutadapt can return one of five modules (which = 1:5). When calling by name the possible modules are summary, adapter1, adapter2, adapter3 or overview. Note that adapter2/3 may be missing from these files depending on the nature of your data. If cutadapt log files are obtained using report=minimal, all supplied log files must be of this format and no modules can be returned.

duplicationMetrics will return either the metrics of histogram. These can be requested by setting which as 1 or 2, or naming either module.

## Value

A tibble. Column names are broadly similar to the text in supplied files, but have been modified for easier handling under R naming conventions.

**Examples**

```
f <- c("bowtiePE.txt", "bowtieSE.txt")
bowtieLogs <- system.file("extdata", f, package = "ngsReports")
df <- importNgsLogs(bowtieLogs, type = "bowtie")
```

importSJ

*Import STAR Splice Junctions***Description**

Import the SJ.out.tab files produced by STAR

**Usage**

```
importSJ(x, stripPaths = TRUE)
```

**Arguments**

**x** vector of file paths to SJ.out.tab files

**stripPaths** logical(1) Remove directory prefixes from the file paths in x

**Details**

Imports one or more splice-junction output files as produced by STAR. If all are located in separated directories with identical names, be sure to set the argument `stripPaths = FALSE`

All co-ordinates are 1-based, in keeping with the STAR manual

**Value**

A tibble

**Author(s)**

Stephen Pederson <stephen.pederson@adelaide.edu.au>

**Examples**

```
sjFiles <- system.file("extdata", "SJ.out.tab", package = "ngsReports")
# Import leaving the complete file path in the column Filename
# The argument stripPaths is set as TRUE by default
df <- importSJ(sjFiles, stripPaths = FALSE)
```

---

isCompressed	<i>Check to see if a file is compressed</i>
--------------	---------------------------------------------

---

### Description

Check to see if a file, or vector of files is compressed

### Usage

```
isCompressed(path, type = c("zip", "gzip"), verbose = FALSE)
```

### Arguments

path	The path to one or more files
type	The type of compression to check for. Currently only ZIP/GZIP files have been implemented.
verbose	logical/integer Determine the level of output to show as messages

### Details

Reads the first four bytes from the local file header. If the file is a .ZIP file, this should match the magic number PK\003\004.

This function assumes that the first thing in a zip archive is the .ZIP entry with the local file header signature. ZIP files containing a self-extracting archive may not exhibit this structure and will return FALSE

### Value

A logical vector

### Examples

```
# Get the files included with the package
fileDir <- system.file("extdata", package = "ngsReports")
allFiles <- list.files(fileDir, pattern = "zip$", full.names = TRUE)
isCompressed(allFiles)
```

---

maxAdapterContent	<i>Get the maximum Adapter Content</i>
-------------------	----------------------------------------

---

### Description

Get the maximum Adapter Content across one or more FASTQC reports

### Usage

```
maxAdapterContent(x, asPercent = TRUE)
```



**Arguments**

`x` Can be a `.FastqcFile`, `FastqcData`, `FastqcDataList` or path

`asPercent` logical. Format the values as percentages with the added % symbol

**Details**

This will extract the `Adapter_Content` module from the supplied object, and provide a tibble with the final value for each file.

**Value**

A tibble object containing the percent of reads with each adapter type at the final position

**Examples**

```
# Get the files included with the package
packageDir <- system.file("extdata", package = "ngsReports")
fl <- list.files(packageDir, pattern = "fastqc.zip", full.names = TRUE)

# Load the FASTQC data as a FastqcDataList object
fdl <- FastqcDataList(fl)

# Get the maxAdapterContent
maxAdapterContent(fdl)
```

mData

*Extract Metadata for TheoreticalGC objects***Description**

Extract Metadata for TheoreticalGC objects

**Usage**

```
mData(object)

## S4 method for signature 'TheoreticalGC'
mData(object)
```

**Arguments**

`object` An object of class `Theoretical GC`

**Value**

A tibble object

**Examples**

```
mData(gcTheoretical)
```

---

overRep2Fasta	<i>Write fasta of Over-Represented sequences.</i>
---------------	---------------------------------------------------

---

## Description

Output overrepresented sequences to disk in fasta format.

## Usage

```
overRep2Fasta(x, path, n = 10, labels, noAdapters = TRUE, ...)

## S4 method for signature 'ANY'
overRep2Fasta(x, path, n = 10, labels, noAdapters = TRUE, ...)

## S4 method for signature 'FastqcData'
overRep2Fasta(x, path, n = 10, labels, noAdapters = TRUE, ...)

## S4 method for signature 'FastqcDataList'
overRep2Fasta(x, path, n = 10, labels, noAdapters = TRUE, ...)
```

## Arguments

x	Can be a FastqcData or FastqcDataList
path	Path to export the fasta file to. Reverts to a default in the working directory if not supplied
n	The number of sequences to output
labels	An optional named factor of labels for the file names. All filenames must be present in the names. File extensions are dropped by default.
noAdapters	logical. Remove any sequences identified as possible adapters or primers by FastQC
...	Used to pass any alternative patterns to remove from the end of filenames

## Details

Fasta will contain Filename, Possible Source, Percent of total reads

## Value

Exports to a fasta file, and returns the fasta information invisibly

## Examples

```
# Get the files included with the package
packageDir <- system.file("extdata", package = "ngsReports")
fl <- list.files(packageDir, pattern = "fastqc.zip", full.names = TRUE)

# Load the FASTQC data as a FastqcDataList object
fdl <- FastqcDataList(fl)

# Export the top10 Overrepresented Sequences as a single fasta file
```

```
faOut <- file.path(tempdir(), "top10.fa")
overRep2Fasta(fdl, path = faOut)
```

---

path

*Return the File Paths from an object*

---

## Description

Return the File Paths from an object

## Usage

```
## S4 method for signature '.FastqcFile'
path(object)

## S4 method for signature 'FastqcData'
path(object)

## S4 method for signature 'FastqcDataList'
path(object)
```

## Arguments

object                    An object of class .FastqcFile

## Details

Obtains the file.path for objects of multiple classes

## Value

A character vector of the file paths to the underlying FastQC reports

## Examples

```
# Get the files included with the package
packageDir <- system.file("extdata", package = "ngsReports")
fl <- list.files(packageDir, pattern = "fastqc.zip", full.names = TRUE)

# Load the FASTQC data as a FastqcDataList object
fdl <- FastqcDataList(fl)
path(fdl)
```

---

plotAdapterContent	<i>Draw an Adapter Content Plot</i>
--------------------	-------------------------------------

---

## Description

Draw an Adapter Content Plot across one or more FASTQC reports

## Usage

```
plotAdapterContent(  
  x,  
  usePlotly = FALSE,  
  labels,  
  pwfCols,  
  warn = 5,  
  fail = 10,  
  ...  
)  
  
## S4 method for signature 'ANY'  
plotAdapterContent(  
  x,  
  usePlotly = FALSE,  
  labels,  
  pwfCols,  
  warn = 5,  
  fail = 10,  
  ...  
)  
  
## S4 method for signature 'character'  
plotAdapterContent(  
  x,  
  usePlotly = FALSE,  
  labels,  
  pwfCols,  
  warn = 5,  
  fail = 10,  
  ...  
)  
  
## S4 method for signature 'FastqcData'  
plotAdapterContent(  
  x,  
  usePlotly = FALSE,  
  labels,  
  pwfCols,  
  warn = 5,  
  fail = 10,  
  ...  
)
```

```
## S4 method for signature 'FastqcDataList'
plotAdapterContent(
  x,
  usePlotly = FALSE,
  labels,
  pwfCols,
  warn = 5,
  fail = 10,
  plotType = c("heatmap", "line"),
  adapterType = "Total",
  cluster = FALSE,
  dendrogram = FALSE,
  ...
)
```

### Arguments

x	Can be a FastqcData, a FastqcDataList or character vector of file paths
usePlotly	logical. Output as ggplot2 (default) or plotly object.
labels	An optional named vector of labels for the file names. All filenames must be present in the names. File extensions are dropped by default.
pwfCols	Object of class <a href="#">PwfCols</a> containing the colours for PASS/WARN/FAIL
warn, fail	The default values for warn and fail are 5 and 10 respectively (i.e. percentages)
...	Used to pass additional attributes to theme() and between methods
plotType	character. Can only take the values plotType = "heatmap" or plotType = "line"
adapterType	A regular expression matching the adapter(s) to be plotted. To plot all adapters summed, specify adapterType = "Total". This is the default behaviour.
cluster	logical default FALSE. If set to TRUE, fastqc data will be clustered using hierarchical clustering
dendrogram	logical redundant if cluster is FALSE if both cluster and dendrogram are specified as TRUE then the dendrogram will be displayed.

### Details

This extracts the Adapter\_Content module from the supplied object and generates a ggplot2 object, with a set of minimal defaults. The output of this function can be further modified using the standard ggplot2 methods.

When x is a single or FastqcData object line plots will always be drawn for all adapters. Otherwise, users can select line plots or heatmaps. When plotting more than one fastqc file, any undetected adapters will not be shown.

An interactive version of the plot can be made by setting usePlotly as TRUE

### Value

A standard ggplot2 object, or an interactive plotly object

## Examples

```
# Get the files included with the package
packageDir <- system.file("extdata", package = "ngsReports")
fl <- list.files(packageDir, pattern = "fastqc.zip", full.names = TRUE)

# Load the FASTQC data as a FastqcDataList object
fdl <- FastqcDataList(fl)

# The default plot
plotAdapterContent(fdl)

# Also subset the reads to just the R1 files
r1 <- grepl("R1", fqName(fdl))
plotAdapterContent(fdl[r1])

# Plot just the Universal Adapter
# and change the y-axis using ggplot2::scale_y_continuous
plotAdapterContent(fdl, adapterType = "Universal", plotType = "line") +
  facet_wrap(~Filename) +
  guides(colour = FALSE)
```

---

plotAlignmentSummary    *Plot a summary of alignments*

---

## Description

Plot a summary of alignments from a set of log files

## Usage

```
plotAlignmentSummary(
  x,
  type = c("star", "bowtie", "bowtie2", "hisat2"),
  usePlotly = FALSE,
  ...,
  fill = c("red", "yellow", "blue", rgb(0, 0.5, 1))
)
```

## Arguments

<code>x</code>	Paths to one or more alignment log files
<code>type</code>	The aligner used. Can be one of star, bowtie, bowtie2 or hisat2
<code>usePlotly</code>	logical. If TRUE an interactive plot will be generated. If FALSE a ggplot object will be output
<code>...</code>	Used to pass additional attributes to theme() and between methods
<code>fill</code>	Colours used to fill the bars. Passed to scale_fill_manual.

## Details

Loads a set of alignment log files and creates a default plot. Implemented aligners are bowtie, bowtie2, Hisat2 and STAR.

**Value**

A ggplot2 object, or a plotly object

**Examples**

```
f <- c("bowtie2PE.txt", "bowtie2SE.txt")
bowtie2Logs <- system.file("extdata", f, package = "ngsReports")
plotAlignmentSummary(bowtie2Logs, "bowtie2")
```

---

plotAssemblyStats	<i>Plot a summary of assembly logs</i>
-------------------	----------------------------------------

---

**Description**

Plot a summary of assembly stats from a set of log files

**Usage**

```
plotAssemblyStats(
  x,
  type = c("quast", "busco"),
  usePlotly = FALSE,
  plotType = c("bar", "paracoord"),
  ...
)
```

**Arguments**

x	Paths to one or more log files
type	The tool used. Can be one of quast or busco
usePlotly	logical. If TRUE an interactive plot will be generated. If FALSE a ggplot object will be output
plotType	character. Plot type to output, one of bar or paracoord.
...	Used to pass additional attributes to theme() and between methods

**Details**

Loads a set of assembly log files and creates a default plot. Implemented tools are quast and BUSCO. quast will plot a parallel coordinate plot of some assembly statistics BUSCO will plot a stacked barplot of completeness statistics

**Value**

A ggplot2 object, or a plotly object

## Examples

```
#getquast log filenames
quastFiles <- system.file("extdata",
  c("quast1.tsv", "quast2.tsv"), package = "ngsReports")

# The default plot
plotAssemblyStats(quastFiles)
```

---

plotBaseQuals	<i>Plot the Base Qualities for each file</i>
---------------	----------------------------------------------

---

## Description

Plot the Base Qualities for each file as separate plots

## Usage

```
plotBaseQuals(
  x,
  usePlotly = FALSE,
  labels,
  pwfCols,
  warn = 25,
  fail = 20,
  boxWidth = 0.8,
  ...
)

## S4 method for signature 'ANY'
plotBaseQuals(
  x,
  usePlotly = FALSE,
  labels,
  pwfCols,
  warn = 25,
  fail = 20,
  boxWidth = 0.8,
  ...
)

## S4 method for signature 'character'
plotBaseQuals(
  x,
  usePlotly = FALSE,
  labels,
  pwfCols,
  warn = 25,
  fail = 20,
  boxWidth = 0.8,
```



```

    ...
)

## S4 method for signature 'FastqcData'
plotBaseQuals(
  x,
  usePlotly = FALSE,
  labels,
  pwfCols,
  warn = 25,
  fail = 20,
  boxWidth = 0.8,
  ...
)

## S4 method for signature 'FastqcDataList'
plotBaseQuals(
  x,
  usePlotly = FALSE,
  labels,
  pwfCols,
  warn = 25,
  fail = 20,
  boxWidth = 0.8,
  plotType = c("heatmap", "boxplot"),
  plotValue = "Mean",
  cluster = FALSE,
  dendrogram = FALSE,
  nc = 2,
  ...
)

```

### Arguments

x	Can be a FastqcData, FastqcDataList or character vector of file paths
usePlotly	logical Default FALSE will render using ggplot. If TRUE plot will be rendered with plotly
labels	An optional named vector of labels for the file names. All filenames must be present in the names. File extensions are dropped by default.
pwfCols	Object of class <a href="#">PwfCols</a> to give colours for pass, warning, and fail values in plot
warn, fail	The default values for warn and fail are 30 and 20 respectively (i.e. percentages)
boxWidth	set the width of boxes when using a boxplot
...	Used to pass additional attributes to theme() and between methods
plotType	character Can be either "boxplot" or "heatmap"
plotValue	character Type of data to be presented. Can be any of the columns returned by getModule(x,module = "Per_base_sequence_qual")
cluster	logical default FALSE. If set to TRUE, fastqc data will be clustered using hierarchical clustering
dendrogram	logical redundant if cluster is FALSE if both cluster and dendrogram are specified as TRUE then the dendrogram will be displayed.

**nc** numeric. The number of columns to create in the plot layout. Only used if drawing boxplots for multiple files in a FastqcDataList

### Details

When acting on a FastqcDataList, this defaults to a heatmap using the mean Per\_base\_sequence\_quality score. A set of plots which replicate those obtained through a standard FastQC html report can be obtained by setting plotType = "boxplot", which uses facet\_wrap to provide the layout as a single ggplot object.

When acting on a FastqcData object, this replicates the Per base sequence quality plots from FastQC with no faceting.

For large datasets, subsetting by R1 or R2 reads may be helpful.

An interactive plot can be obtained by setting usePlotly = TRUE.

### Value

A standard ggplot2 object or an interactive plotly object

### Examples

```
# Get the files included with the package
packageDir <- system.file("extdata", package = "ngsReports")
fl <- list.files(packageDir, pattern = "fastqc.zip", full.names = TRUE)

# Load the FASTQC data as a FastqcDataList object
fdl <- FastqcDataList(fl)

# The default plot for multiple libraries is a heatmap
plotBaseQuals(fdl)

# The default plot for a single library is the standard boxplot
plotBaseQuals(fdl[[1]])
```

---

plotDupLevels	<i>Plot the combined Sequence_Duplication_Levels information</i>
---------------	------------------------------------------------------------------

---

### Description

Plot the Sequence\_Duplication\_Levels information for a set of FASTQC reports

### Usage

```
plotDupLevels(x, usePlotly = FALSE, labels, pwfCols, ...)

## S4 method for signature 'ANY'
plotDupLevels(x, usePlotly = FALSE, labels, pwfCols, ...)

## S4 method for signature 'character'
plotDupLevels(x, usePlotly = FALSE, labels, pwfCols, ...)
```

```
## S4 method for signature 'FastqcData'
plotDupLevels(
  x,
  usePlotly = FALSE,
  labels,
  pwfCols,
  warn = 20,
  fail = 50,
  lineCols = c("red", "blue"),
  ...
)

## S4 method for signature 'FastqcDataList'
plotDupLevels(
  x,
  usePlotly = FALSE,
  labels,
  pwfCols,
  warn = 20,
  fail = 50,
  deduplication = c("pre", "post"),
  plotType = c("heatmap", "line"),
  cluster = FALSE,
  dendrogram = FALSE,
  heatCol = inferno(50),
  ...
)
```

### Arguments

x	Can be a FastqcData, FastqcDataList or file path
usePlotly	logical Default FALSE will render using ggplot. If TRUE plot will be rendered with plotly
labels	An optional named vector of labels for the file names. All filenames must be present in the names. File extensions are dropped by default.
pwfCols	Object of class <a href="#">PwfCols</a> to give colours for pass, warning, and fail values in the plot
...	Used to pass additional attributes to theme() and between methods
warn, fail	The default values for warn and fail are 20 and 50 respectively (i.e. percentages)
lineCols	Colours of the lines drawn for individual libraries
deduplication	Plot Duplication levels 'pre' or 'post' deduplication. Can only take values "pre" and "post"
plotType	Choose between "heatmap" and "line"
cluster	logical default FALSE. If set to TRUE, fastqc data will be clustered using hierarchical clustering
dendrogram	logical redundant if cluster is FALSE if both cluster and dendrogram are specified as TRUE then the dendrogram will be displayed.
heatCol	Colour palette used for the heatmap

## Details

This extracts the `Sequence_Duplication_Levels` from the supplied object and generates a `ggplot2` object, with a set of minimal defaults. For multiple reports, this defaults to a heatmap with block sizes proportional to the percentage of reads belonging to that duplication category.

If setting `usePlotly = FALSE`, the output of this function can be further modified using standard `ggplot2` syntax. If setting `usePlotly = TRUE` an interactive `plotly` object will be produced.

## Value

A standard `ggplot2` or `plotly` object

## Examples

```
# Get the files included with the package
packageDir <- system.file("extdata", package = "ngsReports")
fl <- list.files(packageDir, pattern = "fastqc.zip", full.names = TRUE)

# Load the FASTQC data as a FastqcDataList object
fdl <- FastqcDataList(fl)

# Draw the default plot for a single file
plotDupLevels(fdl[[1]])

plotDupLevels(fdl)
```

---

plotFastqcPCA

*Draw a PCA plot for Fast QC modules*

---

## Description

Draw a PCA plot for Fast QC modules across multiple samples **Experimental**

## Usage

```
plotFastqcPCA(
  x,
  module,
  usePlotly = FALSE,
  labels,
  cluster = FALSE,
  clusterType = "colour",
  groups = NULL,
  ...
)

## S4 method for signature 'ANY'
plotFastqcPCA(
  x,
  module,
  usePlotly = FALSE,
```

```

    labels,
    cluster = FALSE,
    clusterType = "colour",
    groups = NULL,
    ...
)

## S4 method for signature 'character'
plotFastqcPCA(
  x,
  module,
  usePlotly = FALSE,
  labels,
  cluster = FALSE,
  clusterType = "colour",
  groups = NULL,
  ...
)

## S4 method for signature 'FastqcDataList'
plotFastqcPCA(
  x,
  module,
  usePlotly = FALSE,
  labels,
  cluster = FALSE,
  clusterType = "colour",
  groups = NULL,
  ...
)

```

### Arguments

x	Can be a FastqcData, FastqcDataList or file paths
module	character vector containing the desired FastQC module (eg. c("Per_base_sequence_quality", "Per_base_sequence_content"))
usePlotly	logical. Output as ggplot2 (default) or plotly object.
labels	An optional named vector of labels for the file names. All filenames must be present in the names. File extensions are dropped by default
cluster	logical default FALSE. If groups argument is not set fastqc data will be clustered using hierarchical clustering.
clusterType	One of "color/colour" or "hulls". Default is "colours" and will colour points based on cluster/group, "hulls" will draw a polygon around each cluster.
groups	Optional data.frame (or tibble) with columns Filename and Group. Values in the Filename column should correspond to the values returned by fqName(x). If not supplied and cluster = TRUE, clusters will be automatically generated using HCPC from FactoMiner
...	Used to pass additional attributes to theme() and between methods

**Details**

This carries out PCA on all or a subset of FastQC modules and plots the output using either ggplot or plotly. Clustering of the PCA can be carried out using a hierarchical clustering approach. Current modules for PCA are Per\_base\_sequence\_quality, Per\_sequence\_quality\_scores, Per\_sequence\_GC\_content, Per\_base\_sequence\_content, and Sequence\_Length\_Distribution.

**Value**

A standard ggplot2 object, or an interactive plotly object

**Examples**

```
# Get the files included with the package
packageDir <- system.file("extdata", package = "ngsReports")
fl <- list.files(packageDir, pattern = "fastqc.zip", full.names = TRUE)

# Load the FASTQC data as a FastqcDataList object
fdl <- FastqcDataList(fl)
plotFastqcPCA(fdl, module = "Per_sequence_quality_scores", cluster = TRUE)
```

---

plotGcContent

---

*Plot the Per Sequence GC Content*


---

**Description**

Plot the Per Sequence GC Content for a set of FASTQC files

**Usage**

```
plotGcContent(
  x,
  usePlotly = FALSE,
  labels,
  theoreticalGC = TRUE,
  gcType = c("Genome", "Transcriptome"),
  species = "Hsapiens",
  GCOBJECT,
  Fastafilename,
  n = 1e+06,
  ...
)

## S4 method for signature 'ANY'
plotGcContent(
  x,
  usePlotly = FALSE,
  labels,
  theoreticalGC = TRUE,
  gcType = c("Genome", "Transcriptome"),
```

```

    species = "Hsapiens",
    GCOBJECT,
    Fastafilename,
    n = 1e+06,
    ...
)

## S4 method for signature 'character'
plotGcContent(
  x,
  usePlotly = FALSE,
  labels,
  theoreticalGC = TRUE,
  gcType = c("Genome", "Transcriptome"),
  species = "Hsapiens",
  GCOBJECT,
  Fastafilename,
  n = 1e+06,
  ...
)

## S4 method for signature 'FastqcData'
plotGcContent(
  x,
  usePlotly = FALSE,
  labels,
  theoreticalGC = TRUE,
  gcType = c("Genome", "Transcriptome"),
  species = "Hsapiens",
  GCOBJECT,
  Fastafilename,
  n = 1e+06,
  counts = FALSE,
  lineCols = c("red", "blue"),
  ...
)

## S4 method for signature 'FastqcDataList'
plotGcContent(
  x,
  usePlotly = FALSE,
  labels,
  theoreticalGC = TRUE,
  gcType = c("Genome", "Transcriptome"),
  species = "Hsapiens",
  GCOBJECT,
  Fastafilename,
  n = 1e+06,
  plotType = c("heatmap", "line", "cdf"),
  pwfCols,
  cluster = FALSE,
  dendrogram = FALSE,

```

```
    ...  
  )
```

### Arguments

<code>x</code>	Can be a <code>FastqcData</code> , <code>FastqcDataList</code> or character vector of file paths
<code>usePlotly</code>	logical Default FALSE will render using ggplot. If TRUE plot will be rendered with plotly
<code>labels</code>	An optional named vector of labels for the file names. All filenames must be present in the names. File extensions are dropped by default.
<code>theoreticalGC</code>	logical default is FALSE to give the true GC content, set to TRUE to normalize values of <code>GC_Content</code> by the theoretical values using <a href="#">gcTheoretical</a> . species must be specified.
<code>gcType</code>	character Select type of data to normalize GC content against. Accepts either "Genome" (default) or "Transcriptome".
<code>species</code>	character if <code>gcTheory</code> is TRUE it must be accompanied by a species. Species currently supported can be obtained using <code>mData(gcTheoretical)</code>
<code>GCobject</code>	an object of class <code>GCTheoretical</code> . Defaults to the <code>gcTheoretical</code> object supplied with the package
<code>Fastafile</code>	a fasta file contains DNA sequences to generate theoretical GC content
<code>n</code>	number of simulated reads to generate theoretical GC content from <code>Fastafile</code>
<code>...</code>	Used to pass various potting parameters to theme.
<code>counts</code>	logical. Plot the counts from each file if <code>counts = TRUE</code> , otherwise frequencies will be plotted. Ignored if calling the function on a <code>FastqcDataList</code> .
<code>lineCols</code>	Colors for observed and theoretical GC lines in single plots
<code>plotType</code>	Takes values "line", "heatmap" or "cdf"
<code>pwfCols</code>	Object of class <a href="#">PwfCols</a> to give colours for pass, warning, and fail values in plot
<code>cluster</code>	logical default FALSE. If set to TRUE, fastqc data will be clustered using hierarchical clustering
<code>dendrogram</code>	logical redundant if <code>cluster</code> is FALSE if both <code>cluster</code> and <code>dendrogram</code> are specified as TRUE then the dendrogram will be displayed.

### Details

Makes plots for `GC_Content`. When applied to a single `FastqcData` object a simple line plot will be drawn, with Theoretical GC content overlaid if desired.

When applied to multiple FastQC reports, the density at each GC content bin can be shown as a heatmap by setting `theoreticalGC = FALSE`. By default the difference in observed and expected theoretical GC is shown. Species and genome/transcriptome should also be set if utilising the theoretical GC content.

As an alternative to a heatmap, a series of overlaid distributions can be shown by setting `plotType = "line"`.

Can produce a static ggplot2 object or an interactive plotly object.

### Value

A ggplot2 or plotly object



**Examples**

```
# Get the files included with the package
packageDir <- system.file("extdata", package = "ngsReports")
fl <- list.files(packageDir, pattern = "fastqc.zip", full.names = TRUE)

# Load the FASTQC data as a FastqcDataList object
fdl <- FastqcDataList(fl)

# The default plot for a FastqcDataList
plotGcContent(fdl)

# Plot a single FastqcData object
plotGcContent(fdl[[1]])
```

---

plotKmers

*Plot Overrepresented Kmers*


---

**Description**

Plot Overrepresented Kmers

**Usage**

```
plotKmers(x, usePlotly = FALSE, labels, ...)

## S4 method for signature 'ANY'
plotKmers(x, usePlotly = FALSE, labels, ...)

## S4 method for signature 'character'
plotKmers(x, usePlotly = FALSE, labels, ...)

## S4 method for signature 'FastqcData'
plotKmers(
  x,
  usePlotly = FALSE,
  labels,
  n = 6,
  ...,
  lineWidth = 0.5,
  pal = c("red", "blue", "green", "black", "magenta", "yellow")
)

## S4 method for signature 'FastqcDataList'
plotKmers(
  x,
  usePlotly = FALSE,
  labels,
  cluster = FALSE,
  dendrogram = FALSE,
  pwfCols,
```

```

    heatCol = inferno(50),
    ...
  )

```

### Arguments

<code>x</code>	Can be a <code>FastqcData</code> , <code>FastqcDataList</code> or file paths
<code>usePlotly</code>	logical Default FALSE will render using ggplot. If TRUE plot will be rendered with plotly
<code>labels</code>	An optional named vector of labels for the file names. All filenames must be present in the names. File extensions are dropped by default.
<code>...</code>	Used to pass various potting parameters to theme. Can also be used to set size and colour for box outlines.
<code>n</code>	numeric. The number of Kmers to show.
<code>lineWidth</code>	Passed to <code>geom_line(size = lineWidth)</code>
<code>pal</code>	The colour palette. If the vector supplied is less than <code>n</code> , <code>grDevices::colorRampPalette()</code> will be used
<code>cluster</code>	logical default FALSE. If set to TRUE, fastqc data will be clustered using hierarchical clustering
<code>dendrogram</code>	logical redundant if <code>cluster</code> is FALSE if both <code>cluster</code> and <code>dendrogram</code> are specified as TRUE then the dendrogram will be displayed.
<code>pwfCols</code>	Object of class <code>PwfCols</code> to give colours for pass, warning, and fail values in the plot
<code>heatCol</code>	Colour palette used for the heatmap. Default is <code>inferno</code> from the package <code>viridis</code>

### Details

As the Kmer Content module present in FastQC reports is relatively uninformative, and omitted by default in later versions of FastQC, these are rudimentary plots.

Plots for `FastqcData` objects replicate those contained in a FastQC report, whilst the heatmap generated from `FastqcDataList` objects simply show the location and abundance of over-represented Kmers.

### Value

A standard `ggplot2` object or an interactive `plotly` object

### Examples

```

# Get the files included with the package
packageDir <- system.file("extdata", package = "ngsReports")
f1 <- list.files(packageDir, pattern = "fastqc.zip", full.names = TRUE)

# Load the FASTQC data as a FastqcDataList object
fdl <- FastqcDataList(f1)
plotKmers(fdl[[1]])

```

plotNContent

*Draw an N Content Plot***Description**

Draw an N Content Plot across one or more FastQC reports

**Usage**

```
plotNContent(x, usePlotly = FALSE, labels, pwfCols, warn = 5, fail = 20, ...)
```

```
## S4 method for signature 'ANY'
```

```
plotNContent(x, usePlotly = FALSE, labels, pwfCols, warn = 5, fail = 20, ...)
```

```
## S4 method for signature 'character'
```

```
plotNContent(x, usePlotly = FALSE, labels, pwfCols, warn = 5, fail = 20, ...)
```

```
## S4 method for signature 'FastqcData'
```

```
plotNContent(
  x,
  usePlotly = FALSE,
  labels,
  pwfCols,
  warn = 5,
  fail = 20,
  ...,
  lineCol = "red"
)
```

```
## S4 method for signature 'FastqcDataList'
```

```
plotNContent(
  x,
  usePlotly = FALSE,
  labels,
  pwfCols,
  warn = 5,
  fail = 20,
  cluster = FALSE,
  dendrogram = FALSE,
  ...
)
```

**Arguments**

x	Can be a FastqcData, FastqcDataList or file paths
usePlotly	logical. Output as ggplot2 (default) or plotly object.
labels	An optional named vector of labels for the file names. All filenames must be present in the names. File extensions are dropped by default
pwfCols	Object of class <a href="#">PwfCols</a> containing the colours for PASS/WARN/FAIL
warn, fail	The default values for warn and fail are 5 and 10 respectively (i.e. percentages)

...	Used to pass additional attributes to theme() and between methods
lineCol	Defaults to red
cluster	logical default FALSE. If set to TRUE, fastqc data will be clustered using hierarchical clustering
dendrogram	logical redundant if cluster is FALSE if both cluster and dendrogram are specified as TRUE then the dendrogram will be displayed.

### Details

This extracts the N\_Content from the supplied object and generates a ggplot2 object, with a set of minimal defaults. The output of this function can be further modified using the standard ggplot2 methods.

When x is a single FastqcData object line plots will always be drawn for all Ns. Otherwise, users can select line plots or heatmaps.

### Value

A standard ggplot2 object, or an interactive plotly object

### Examples

```
# Get the files included with the package
packageDir <- system.file("extdata", package = "ngsReports")
fl <- list.files(packageDir, pattern = "fastqc.zip", full.names = TRUE)

# Load the FASTQC data as a FastqcDataList object
fdl <- FastqcDataList(fl)

# The default plot
plotNContent(fdl[[1]])
```

---

plotOverrep

*Plot a summary of Over-represented Sequences*

---

### Description

Plot a summary of Over-represented Sequences for a set of FASTQC reports

### Usage

```
plotOverrep(x, usePlotly = FALSE, labels, pwfCols, ...)

## S4 method for signature 'ANY'
plotOverrep(x, usePlotly = FALSE, labels, pwfCols, ...)

## S4 method for signature 'character'
plotOverrep(x, usePlotly = FALSE, labels, pwfCols, ...)

## S4 method for signature 'FastqcData'
```

```

plotOverrep(
  x,
  usePlotly = FALSE,
  labels,
  pwfCols,
  n = 10,
  ...,
  expand.x = expansion(mult = c(0, 0.05)),
  expand.y = expansion(0, 0.6)
)

## S4 method for signature 'FastqcDataList'
plotOverrep(
  x,
  usePlotly = FALSE,
  labels,
  pwfCols,
  cluster = FALSE,
  dendrogram = FALSE,
  ...,
  paletteName = "Set1",
  expand.x = expansion(mult = c(0, 0.05)),
  expand.y = expansion(0, 0)
)

```

### Arguments

<code>x</code>	Can be a <code>FastqcData</code> , <code>FastqcDataList</code> or file paths
<code>usePlotly</code>	logical Default FALSE will render using ggplot. If TRUE plot will be rendered with plotly
<code>labels</code>	An optional named factor of labels for the file names. All filenames must be present in the names. File extensions are dropped by default.
<code>pwfCols</code>	Object of class <code>PwfCols</code> containing the colours for PASS/WARN/FAIL
<code>...</code>	Used to pass additional attributes to <code>theme()</code> and between methods
<code>n</code>	The number of sequences to plot from an individual file
<code>expand.x</code> , <code>expand.y</code>	Output from <code>expansion()</code> or numeric vectors of length 4. Passed to <code>scale_*_continuous()</code>
<code>cluster</code>	logical default FALSE. If set to TRUE, fastqc data will be clustered using hierarchical clustering
<code>dendrogram</code>	logical redundant if <code>cluster</code> is FALSE if both <code>cluster</code> and <code>dendrogram</code> are specified as TRUE then the dendrogram will be displayed.
<code>paletteName</code>	Name of the palette for colouring the possible sources of the overrepresented sequences. Must be a palette name from <code>RColorBrewer</code>

### Details

Percentages are obtained by simply summing those within a report. Any possible double counting by FastQC is ignored for the purposes of a simple approximation.

Plots generated from a `FastqcData` object will show the top `n` sequences grouped by their predicted source & coloured by whether the individual sequence would cause a WARN/FAIL.

Plots generated from a FastqcDataList group sequences by predicted source and summarise as a percentage of the total reads.

### Value

A standard ggplot2 object

### Examples

```
# Get the files included with the package
packageDir <- system.file("extdata", package = "ngsReports")
fl <- list.files(packageDir, pattern = "fastqc.zip", full.names = TRUE)

# Load the FASTQC data as a FastqcDataList object
fdl <- FastqcDataList(fl)

# Another example which isn't ideal
plotOverrep(fdl)
```

---

plotReadTotals	<i>Draw a barplot of read totals</i>
----------------	--------------------------------------

---

### Description

Draw a barplot of read totals

### Usage

```
plotReadTotals(
  x,
  usePlotly = FALSE,
  labels,
  duplicated = TRUE,
  bars = c("stacked", "adjacent"),
  barCols = c("red", "blue"),
  expand.x = expansion(mult = c(0, 0.02)),
  ...
)

## S4 method for signature 'ANY'
plotReadTotals(
  x,
  usePlotly = FALSE,
  labels,
  duplicated = TRUE,
  bars = c("stacked", "adjacent"),
  barCols = c("red", "blue"),
  expand.x = expansion(mult = c(0, 0.02)),
  ...
)
```

```
## S4 method for signature 'character'
plotReadTotals(
  x,
  usePlotly = FALSE,
  labels,
  duplicated = TRUE,
  bars = c("stacked", "adjacent"),
  barCols = c("red", "blue"),
  expand.x = expansion(mult = c(0, 0.02)),
  ...
)

## S4 method for signature 'FastqcDataList'
plotReadTotals(
  x,
  usePlotly = FALSE,
  labels,
  duplicated = TRUE,
  bars = c("stacked", "adjacent"),
  barCols = c("red", "blue"),
  expand.x = expansion(mult = c(0, 0.02)),
  ...
)
```

### Arguments

<code>x</code>	Can be a <code>FastqcData</code> , <code>FastqcDataList</code> or file paths
<code>usePlotly</code>	logical Default FALSE will render using <code>ggplot</code> . If TRUE plot will be rendered with <code>plotly</code>
<code>labels</code>	An optional named vector of labels for the file names. All filenames must be present in the names. File extensions are dropped by default.
<code>duplicated</code>	logical. Include deduplicated read total estimates to plot charts
<code>bars</code>	If <code>duplicated = TRUE</code> , show unique and deduplicated reads as "stacked" or "adjacent".
<code>barCols</code>	Colours for duplicated and unique reads.
<code>expand.x</code>	Output from <code>expansion()</code> controlling x-axis expansion. Alternatively can be a numeric vector of length 4
<code>...</code>	Used to pass additional attributes to <code>theme()</code>

### Details

Draw a barplot of read totals using the standard `ggplot2` syntax. The raw data from `readTotals` can otherwise be used to manually create a plot.

Duplication levels are based on the value shown on FASTQC reports at the top of the DeDuplicated-Totals plot, which is known to be inaccurate. As it still gives a good guide as to sequence diversity it is included as the default. This can be turned off by setting `duplicated = FALSE`.

### Value

Returns a `ggplot` or `plotly` object

## Examples

```
# Get the files included with the package
packageDir <- system.file("extdata", package = "ngsReports")
fl <- list.files(packageDir, pattern = "fastqc.zip", full.names = TRUE)

# Load the FASTQC data as a FastqcDataList object
fdl <- FastqcDataList(fl)

# Plot the Read Totals showing estimated duplicates
plotReadTotals(fdl)

# Plot the Read Totals without estimated duplicates
plotReadTotals(fdl, duplicated = FALSE)
```

---

plotSeqContent	<i>Plot the per base content as a heatmap</i>
----------------	-----------------------------------------------

---

## Description

Plot the Per Base content for a set of FASTQC files.

## Usage

```
plotSeqContent(x, usePlotly = FALSE, labels, ...)

## S4 method for signature 'ANY'
plotSeqContent(x, usePlotly = FALSE, labels, ...)

## S4 method for signature 'character'
plotSeqContent(x, usePlotly = FALSE, labels, ...)

## S4 method for signature 'FastqcData'
plotSeqContent(x, usePlotly = FALSE, labels, ...)

## S4 method for signature 'FastqcDataList'
plotSeqContent(
  x,
  usePlotly = FALSE,
  labels,
  pwfCols,
  plotType = c("heatmap", "line", "residuals"),
  cluster = FALSE,
  dendrogram = FALSE,
  ...,
  nc = 2
)
```



**Arguments**

<code>x</code>	Can be a <code>FastqcData</code> , <code>FastqcDataList</code> or file paths
<code>usePlotly</code>	logical. Generate an interactive plot using plotly
<code>labels</code>	An optional named vector of labels for the file names. All file names must be present in the names of the vector. File extensions are dropped by default.
<code>...</code>	Used to pass additional attributes to <code>theme()</code> and between methods
<code>pwfCols</code>	Object of class <code>PwfCols</code> to give colours for pass, warning, and fail values in plot
<code>plotType</code>	character. Type of plot to generate. Must be "line", "heatmap" or "residuals"
<code>cluster</code>	logical default FALSE. If set to TRUE, fastqc data will be clustered using hierarchical clustering
<code>dendrogram</code>	logical redundant if <code>cluster</code> is FALSE if both <code>cluster</code> and <code>dendrogram</code> are specified as TRUE then the dendrogram will be displayed.
<code>nc</code>	Specify the number of columns if plotting a <code>FastqcDataList</code> as line plots. Passed to <code>ggplot2::facet_wrap</code> .

**Details**

Per base sequence content ( heatmap colours when plotting from multiple reports. The individual line plots are able to be generated by setting `plotType = "line"`, and the layout is determined by `facet_wrap` from `ggplot2`.

Individual line plots are also generated when plotting from a single `FastqcData` object.

If setting `usePlotly = TRUE` for a large number of reports, the plot can be slow to render. An alternative may be to produce a plot of residuals for each base, produced by taking the position-specific mean for each base.

**Value**

A `ggplot2` object or an interactive `plotly` object

**Examples**

```
# Get the files included with the package
packageDir <- system.file("extdata", package = "ngsReports")
fl <- list.files(packageDir, pattern = "fastqc.zip", full.names = TRUE)

# Load the FASTQC data as a FastqcDataList object
fdl <- FastqcDataList(fl)

# The default plot
plotSeqContent(fdl)
```

---

plotSeqLengthDistn      *Plot the Sequence Length Distribution*

---

## Description

Plot the Sequence Length Distribution across one or more FASTQC reports

## Usage

```
plotSeqLengthDistn(x, usePlotly = FALSE, labels, ...)
```

```
## S4 method for signature 'ANY'
```

```
plotSeqLengthDistn(x, usePlotly = FALSE, labels, ...)
```

```
## S4 method for signature 'character'
```

```
plotSeqLengthDistn(x, usePlotly = FALSE, labels, ...)
```

```
## S4 method for signature 'FastqcData'
```

```
plotSeqLengthDistn(
  x,
  usePlotly = FALSE,
  labels,
  plotType = c("line", "cdf"),
  ...,
  expand.x = expansion(0, 0.2)
)
```

```
## S4 method for signature 'FastqcDataList'
```

```
plotSeqLengthDistn(
  x,
  usePlotly = FALSE,
  labels,
  counts = FALSE,
  plotType = c("heatmap", "line", "cdf"),
  cluster = FALSE,
  dendrogram = FALSE,
  ...,
  expand.x = expansion(0, 0.2),
  heatCol = inferno(50)
)
```

## Arguments

x	Can be a FastqcData, FastqcDataList or file paths
usePlotly	logical. Output as ggplot2 or plotly object.
labels	An optional named vector of labels for the file names. All filenames must be present in the names. File extensions are dropped by default.
...	Used to pass additional attributes to theme()
plotType	character. Can only take the values plotType = "heatmap" plotType = "line" or plotType = "cdf"

expand.x	Output from expansion() or numeric vector of length 4. Passed to scale_x_discrete
counts	logical Should distributions be shown as counts or frequencies (percentages)
cluster	logical default FALSE. If set to TRUE, fastqc data will be clustered using hierarchical clustering
dendrogram	logical redundant if cluster and usePlotly are FALSE. If both cluster and dendrogram are specified as TRUE then the dendrogram will be displayed.
heatCol	The colour scheme for the heatmap

## Details

This extracts the Sequence Length Distribution from the supplied object and generates a ggplot2 object, with a set of minimal defaults. The output of this function can be further modified using the standard ggplot2 methods.

A cdf plot can also be generated to provide guidance for minimum read length in some NGS workflows, by setting plotType = "cdf". If all libraries have reads of identical lengths, these plots may be less informative.

An alternative interactive plot is available by setting the argument usePlotly = TRUE.

## Value

A standard ggplot2 object, or an interactive plotly object

## Examples

```
# Get the files included with the package
packageDir <- system.file("extdata", package = "ngsReports")
fl <- list.files(packageDir, pattern = "fastqc.zip", full.names = TRUE)

# Load the FASTQC data as a FastqcDataList object
fdl <- FastqcDataList(fl)

# Plot as a frequency plot using lines
plotSeqLengthDistn(fdl)

# Or plot the cdf
plotSeqLengthDistn(fdl, plotType = "cdf")
```

---

plotSeqQuals

---

*Plot the Per Sequence Quality Scores*


---

## Description

Plot the Per Sequence Quality Scores for a set of FASTQC reports

**Usage**

```
plotSeqQuals(  
  x,  
  usePlotly = FALSE,  
  labels,  
  pwfCols,  
  counts = FALSE,  
  alpha = 0.1,  
  warn = 30,  
  fail = 20,  
  ...  
)  
  
## S4 method for signature 'ANY'  
plotSeqQuals(  
  x,  
  usePlotly = FALSE,  
  labels,  
  pwfCols,  
  counts = FALSE,  
  alpha = 0.1,  
  warn = 30,  
  fail = 20,  
  ...  
)  
  
## S4 method for signature 'character'  
plotSeqQuals(  
  x,  
  usePlotly = FALSE,  
  labels,  
  pwfCols,  
  counts = FALSE,  
  alpha = 0.1,  
  warn = 30,  
  fail = 20,  
  ...  
)  
  
## S4 method for signature 'FastqcData'  
plotSeqQuals(  
  x,  
  usePlotly = FALSE,  
  labels,  
  pwfCols,  
  counts = FALSE,  
  alpha = 0.1,  
  warn = 30,  
  fail = 20,  
  ...  
)
```

```
## S4 method for signature 'FastqcDataList'
plotSeqQuals(
  x,
  usePlotly = FALSE,
  labels,
  pwfCols,
  counts = FALSE,
  alpha = 0.1,
  warn = 30,
  fail = 20,
  plotType = c("heatmap", "line"),
  dendrogram = FALSE,
  cluster = FALSE,
  ...
)
```

### Arguments

x	Can be a FastqcData, FastqcDataList or path
usePlotly	logical Default FALSE will render using ggplot. If TRUE plot will be rendered with plotly
labels	An optional named factor of labels for the file names. All filenames must be present in the names. File extensions are dropped by default.
pwfCols	Object of class <a href="#">PwfCols</a> containing the colours for PASS/WARN/FAIL
counts	logical. Plot the counts from each file if counts = TRUE, otherwise the frequencies will be plotted
alpha	set alpha for line graph bounds
warn, fail	The default values for warn and fail are 5 and 10 respectively (i.e. percentages)
...	Used to pass various potting parameters to theme. Can also be used to set size and colour for box outlines.
plotType	character. Can only take the values plotType = "heatmap" or plotType = "line"
dendrogram	logical redundant if cluster is FALSE if both cluster and dendrogram are specified as TRUE then the dendrogram will be displayed.
cluster	logical default FALSE. If set to TRUE, fastqc data will be clustered using hierarchical clustering

### Details

Plots the distribution of average sequence quality scores across the set of files. Values can be plotted either as counts (counts = TRUE) or as frequencies (counts = FALSE).

Any faceting or scale adjustment can be performed after generation of the initial plot, using the standard methods of ggplot2 as desired.

### Value

A standard ggplot2 object, or an interactive plotly object

## Examples

```
# Get the files included with the package
packageDir <- system.file("extdata", package = "ngsReports")
fl <- list.files(packageDir, pattern = "fastqc.zip", full.names = TRUE)

# Load the FASTQC data as a FastqcDataList object
fdl <- FastqcDataList(fl)

# The default plot
plotSeqQuals(fdl)

# Also subset the reads to just the R1 files
r1 <- grepl("R1", fqName(fdl))
plotSeqQuals(fdl[r1])
```

---

plotSummary

*Plot the PASS/WARN/FAIL information*

---

## Description

Extract the PASS/WARN/FAIL summaries and plot them

## Usage

```
plotSummary(
  x,
  usePlotly = FALSE,
  labels,
  pwfCols,
  cluster = FALSE,
  dendrogram = FALSE,
  ...
)

## S4 method for signature 'ANY'
plotSummary(
  x,
  usePlotly = FALSE,
  labels,
  pwfCols,
  cluster = FALSE,
  dendrogram = FALSE,
  ...
)

## S4 method for signature 'character'
plotSummary(
  x,
  usePlotly = FALSE,
  labels,
```

```

    pwfCols,
    cluster = FALSE,
    dendrogram = FALSE,
    ...
)

## S4 method for signature 'FastqcDataList'
plotSummary(
  x,
  usePlotly = FALSE,
  labels,
  pwfCols,
  cluster = FALSE,
  dendrogram = FALSE,
  ...,
  gridlineWidth = 0.2,
  gridlineCol = "grey20"
)

```

### Arguments

<code>x</code>	Can be a <code>FastqcData</code> , <code>FastqcDataList</code> or character vector of file paths
<code>usePlotly</code>	logical. Generate an interactive plot using plotly
<code>labels</code>	An optional named vector of labels for the file names. All filenames must be present in the names. File extensions are dropped by default.
<code>pwfCols</code>	Object of class <code>PwfCols</code> containing the colours for PASS/WARN/FAIL
<code>cluster</code>	logical default FALSE. If set to TRUE, fastqc data will be clustered using hierarchical clustering
<code>dendrogram</code>	logical redundant if cluster is FALSE if both cluster and dendrogram are specified as TRUE then the dendrogram will be displayed.
<code>...</code>	Used to pass various potting parameters to theme.
<code>gridlineWidth</code> , <code>gridlineCol</code>	Passed to <code>geom_hline</code> and <code>geom_vline</code> to determine width and colour of grid-lines

### Details

This uses the standard ggplot2 syntax to create a three colour plot. The output of this function can be further modified using the standard ggplot2 methods if required.

### Value

A ggplot2 object (`usePlotly = FALSE`) or an interactive plotly object (`usePlotly = TRUE`)

### Examples

```

# Get the files included with the package
packageDir <- system.file("extdata", package = "ngsReports")
fl <- list.files(packageDir, pattern = "fastqc.zip", full.names = TRUE)

# Load the FASTQC data as a FastqcDataList object

```

```
fdl <- FastqcDataList(f1)

# Check the overall PASS/WARN/FAIL status
plotSummary(fdl)
```

---

pwf

*Colours for PASS/WARN/FAIL*


---

### Description

Default colours for PASS/WARN/FAIL

### Usage

pwf

### Format

An object of class PwfCols of length 1.

### Details

pwf is an object of class PwfCols supplied with the package and used as the default colouring. Colours correspond approximately to PASS, WARN and FAIL from the FASTQC reports, with the additional colour (MAX) included to indicate an extreme FAIL. In order, these colours in the default vector are green (`rgb(0,0.8,0)`), yellow (`rgb(0.9,0.9,0.2)`), red (`rgb(0.8,0.2,0.2)`) and white (`rgb(1,1,1)`)

### Examples

```
# Make a pie chart showing the default colours
pie(rep(1,4), labels = names(pwf), col = getColours(pwf))
```

---

PwfCols-class

*The PwfCols class and associated methods*


---

### Description

Define the PwfCols class and associated methods

### Details

This is an object of with four colours in components named PASS, WARN, FAIL and MAX. Used to indicate these categories as defined on the standard plots from fastqc.



**Slots**

PASS A vector of length 1, defining the colour for PASS in rgb format. Defaults to rgb(0, 0.8, 0)

WARN A vector of length 1, defining the colour for WARN in rgb format. Defaults to rgb(0.9, 0.9, 0.2)

FAIL A vector of length 1, defining the colour for FAIL in rgb format. Defaults to rgb(0.8, 0.2, 0.2)

MAX A vector of length 1, defining the colour for an extreme FAIL or NA in rgb format. Defaults to rgb(1, 1, 1)

---

readTotals	<i>Get the read totals</i>
------------	----------------------------

---

**Description**

Get the read totals from one or more FASTQC reports

**Usage**

```
readTotals(x)
```

**Arguments**

x Can be a FastqcData, FastqcDataList or file paths

**Value**

A tibble with the columns Filename and Total\_Sequences

**Examples**

```
# Get the files included with the package
packageDir <- system.file("extdata", package = "ngsReports")
fl <- list.files(packageDir, pattern = "fastqc.zip", full.names = TRUE)

# Load the FASTQC data as a FastqcDataList object
fdl <- FastqcDataList(fl)

# Print the read totals
readTotals(fdl)
```

---

`runFastQC`*A wrapper for the bash shell command fastqc.*

---

## Description

**Soft-deprecated** A convenient wrapper for the bash shell command fastqc. Only runs if Fastqc is installed.

## Usage

```
runFastQC(  
  object,  
  outPath,  
  threads = 1L,  
  casava = FALSE,  
  nofilter = FALSE,  
  extract = FALSE,  
  nogroup = FALSE,  
  min_length = 1,  
  contaminants = c(),  
  adapters = c(),  
  kmers = 7,  
  exec  
)  
  
## S4 method for signature 'ANY'  
runFastQC(  
  object,  
  outPath,  
  threads = 1L,  
  casava = FALSE,  
  nofilter = FALSE,  
  extract = FALSE,  
  nogroup = FALSE,  
  min_length = 1,  
  contaminants = c(),  
  adapters = c(),  
  kmers = 7,  
  exec  
)  
  
## S4 method for signature 'FastqFile'  
runFastQC(  
  object,  
  outPath,  
  threads = 1L,  
  casava = FALSE,  
  nofilter = FALSE,  
  extract = FALSE,  
  nogroup = FALSE,  
  min_length = 1,
```

```
        contaminants = c(),
        adapters = c(),
        kmers = 7,
        exec
    )

## S4 method for signature 'FastqFileList'
runFastQC(
    object,
    outPath,
    threads = 1L,
    casava = FALSE,
    nofilter = FALSE,
    extract = FALSE,
    nogroup = FALSE,
    min_length = 1,
    contaminants = c(),
    adapters = c(),
    kmers = 7,
    exec
)

## S4 method for signature 'BamFile'
runFastQC(
    object,
    outPath,
    threads = 1L,
    casava = FALSE,
    nofilter = FALSE,
    extract = FALSE,
    nogroup = FALSE,
    min_length = 1,
    contaminants = c(),
    adapters = c(),
    kmers = 7,
    exec
)

## S4 method for signature 'BamFileList'
runFastQC(
    object,
    outPath,
    threads = 1L,
    casava = FALSE,
    nofilter = FALSE,
    extract = FALSE,
    nogroup = FALSE,
    min_length = 1,
    contaminants = c(),
    adapters = c(),
    kmers = 7,
    exec
)
```

)

**Arguments**

object	A FastqFileList, BamFileList, FastqFile, BamFile or character vector of file paths with all objects coercible to a single one of these types.
outPath	The path to write the FastQC reports. Must exist as (for safety) it will not be created when calling this function
threads	The number of threads to run in parallel
casava	logical. Sets the --casava flag
nofilter	logical. Sets the --nofilter flag
extract	logical. Extract the zip files on completion of the report
nogroup	logical. Sets the grouping of bases for reads longer than 50bp
min_length	integer. Sets an artificial lower limit on the length of the sequence to be shown in the report.
contaminants	Path to an alternate file with contaminants. The structure of the file will not be checked. Refer to the fastqc help page for more details
adapters	Path to a file listing adapters to search for. The structure of the file will not be checked. Refer to the fastqc help page for more details
kmers	An integer between 2 and 10
exec	The location of the fastqc executable.

**Details**

This is a simple wrapper function for controlling & running fastqc from within R. This can be very useful for controlling & documenting an entire pipeline from within knitr to produce a simple report

Takes a FastqFile, FastqFileList, BamFile or BamFileList. Alternatively paths to files which are coercible to these objects can be passed.

Only the common functionality of FastQC is implemented, for more fine detail please call FastQC directly.

**Value**

An list of paths to the output

**Author(s)**

Stephen Pederson <stephen.pederson@adelaide.edu.au>

**Examples**

```
## Not run:
library(ShortRead)
sp <- SolexaPath(system.file('extdata', package='ShortRead'))
fl <- file.path(analysisPath(sp), "s_1_sequence.txt")
f <- FastqFile(fl)
# This requires a working installation of FastQC
fqFile <- runFastQC(f, outPath = tempdir())

## End(Not run)
```

---

TheoreticalGC-class	<i>The TheoreticalGC Object Class</i>
---------------------	---------------------------------------

---

**Description**

Contains Theoretical GC content for a selection of species

**Details**

Estimates are able to be retained for genomic and transcriptomic sequences. Values are stored as frequencies.

**Value**

An object of class TheoreticalGC

**Slots**

Genome A data.frame containing theoretical GC content for genomic sequences

Transcriptome A data.frame containing theoretical GC content for transcriptomic sequences

mData A data.frame containing metadata about all species in the object

**Examples**

```
## How to form an object using your own fasta file
faDir <- system.file("extdata", package = "ngsReports")
faFile <- list.files(faDir, pattern = "fasta", full.names = TRUE)
gen_df <- estGcDistn(faFile, n = 200)
gen_df <- dplyr::rename(gen_df, Athaliana = Freq)
mData_df <-
  data.frame(Name = "Athaliana", Genome = TRUE, Transcriptome = FALSE)
tr_df <- data.frame()
myGC <- new(
  "TheoreticalGC", Genome = gen_df, Transcriptome = tr_df, mData = mData_df)
```

---

writeHtmlReport	<i>Write an HTML Summary Report</i>
-----------------	-------------------------------------

---

**Description**

Compiles an HTML report using a supplied template

**Usage**

```
writeHtmlReport(
  fastqcDir,
  template,
  outDir,
  usePlotly = TRUE,
  species = "Hsapiens",
  gcType = c("Genome", "Transcriptome"),
  nOver = 30,
  targetsDF,
  overwrite = FALSE,
  quiet = TRUE
)
```

**Arguments**

fastqcDir	A directory containing zipped, or extracted FastQC reports
template	The template file which will be copied into fastqcDir
outDir	The directory to write the compiled document to
usePlotly	Generate interactive plots?
species	Species/closely related species of sequenced samples
gcType	Is the data "Transcriptomic" or "Genomic" in nature?
nOver	The maximum number of Overrepresented Sequences to show
targetsDF	A data.frame with at least two columns named Filename and Label. The file-names should match the original fastq files, and the labels should be simply alternative labels for these files for convenience.
overwrite	logical. Overwrite any previous copies of the template file in the destination directory
quiet	logical. Show or hide markdown output in the Console.

**Details**

This will take a user supplied template, or the file supplied with the package and create an HTML summary of all standard FASTQC plots for all files in the supplied directory.

**Value**

Silently returns TRUE and will output a compiled HTML file from the supplied Rmarkdown template file

**Examples**

```
## Not run:
packageDir <- system.file("extdata", package = "ngsReports")
fileList <- list.files(packageDir, pattern = "fastqc.zip", full.names= TRUE)
# Copy these files to tempdir() to avoid overwriting
# any files in the package directory
file.copy(fileList, tempdir(), overwrite = TRUE)
writeHtmlReport(fastqcDir = tempdir())

## End(Not run)
```

---

[ *Extract Elements*


---

**Description**

Extract elements from FastqcDataList Object

**Usage**

```
## S4 method for signature 'FastqcDataList,numeric,missing,ANY'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'FastqcDataList,character,missing,ANY'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'FastqcDataList,logical,missing,ANY'
x[i, j, ..., drop = TRUE]

## S4 method for signature 'FastqcDataList,ANY,missing,ANY'
x[i, j, ..., drop = TRUE]
```

**Arguments**

x	A FastqcDataList
i	character, logical or integer vector
j	not used
...	not used
drop	not used

**Details**

Extract elements in a consistent manner with R conventions

**Value**

Will return a subset of the original object following the standard rules for subsetting objects

**Examples**

```
# Get the files included with the package
packageDir <- system.file("extdata", package = "ngsReports")
f1 <- list.files(packageDir, pattern = "fastqc.zip", full.names = TRUE)

# Load the FASTQC data as a FastqcDataList object
fdl <- FastqcDataList(f1)

# Subsetting using the standard methods
fdl[1]
fdl[[1]]
```

# Index

- \* **datasets**
  - gcTheoretical, [9](#)
  - pwf, [48](#)
- [\[, 55\]](#)
- [\[, FastqcDataList, ANY, missing, ANY-method](#)
  - [\(\[\], 55\)](#)
- [\[, FastqcDataList, character, missing, ANY-method](#)
  - [\(\[\], 55\)](#)
- [\[, FastqcDataList, logical, missing, ANY-method](#)
  - [\(\[\], 55\)](#)
- [\[, FastqcDataList, numeric, missing, ANY-method](#)
  - [\(\[\], 55\)](#)
- estGcDistn, [4](#)
- estGcDistn, ANY-method (estGcDistn), [4](#)
- estGcDistn, character-method
  - (estGcDistn), [4](#)
- estGcDistn, DNAStrngSet-method
  - (estGcDistn), [4](#)
- FastqcData (FastqcData-class), [5](#)
- FastqcData-class, [5](#)
- FastqcDataList (FastqcDataList-class), [6](#)
- FastqcDataList-class, [6](#)
- fqcVersion
  - (fqcVersion, FastqcData-method), [7](#)
- fqcVersion, ANY-method
  - (fqcVersion, FastqcData-method), [7](#)
- fqcVersion, FastqcData-method, [7](#)
- fqcVersion, FastqcDataList-method
  - (fqcVersion, FastqcData-method), [7](#)
- fqName, [8](#)
- fqName, ANY-method (fqName), [8](#)
- fqName, FastqcData-method (fqName), [8](#)
- fqName, FastqcDataList-method (fqName), [8](#)
- gcAvail, [8](#)
- gcAvail, TheoreticalGC-method (gcAvail), [8](#)
- gcTheoretical, [9, 32](#)
- getColours, [10](#)
- getColours, PwfCols-method (getColours), [10](#)
- getGC, [11](#)
- getGC, ANY-method (getGC), [11](#)
- getGC, TheoreticalGC-method (getGC), [11](#)
- getModule, [5](#)
- getModule
  - (getModule, FastqcData-method), [12](#)
- getModule, ANY-method
  - (getModule, FastqcData-method), [12](#)
- getModule, FastqcData-method, [12](#)
- getModule, FastqcDataList-method
  - (getModule, FastqcData-method), [12](#)
- getSummary
  - (getSummary, .FastqcFile-method), [13](#)
- getSummary, .FastqcFile-method, [13](#)
- getSummary, ANY-method
  - (getSummary, .FastqcFile-method), [13](#)
- getSummary, FastqcData-method
  - (getSummary, .FastqcFile-method), [13](#)
- getSummary, FastqcDataList-method
  - (getSummary, .FastqcFile-method), [13](#)
- importNgsLogs, [14](#)
- importSJ, [15](#)
- isCompressed, [16](#)
- maxAdapterContent, [16](#)
- mData, [17](#)
- mData, TheoreticalGC-method (mData), [17](#)
- overRep2Fasta, [18](#)
- overRep2Fasta, ANY-method
  - (overRep2Fasta), [18](#)
- overRep2Fasta, FastqcData-method
  - (overRep2Fasta), [18](#)



- overRep2Fasta, FastqcDataList-method  
(overRep2Fasta), 18
- path, 19
- path, .FastqcFile-method (path), 19
- path, FastqcData-method (path), 19
- path, FastqcDataList-method (path), 19
- plotAdapterContent, 20
- plotAdapterContent, ANY-method  
(plotAdapterContent), 20
- plotAdapterContent, character-method  
(plotAdapterContent), 20
- plotAdapterContent, FastqcData-method  
(plotAdapterContent), 20
- plotAdapterContent, FastqcDataList-method  
(plotAdapterContent), 20
- plotAlignmentSummary, 22
- plotAssemblyStats, 23
- plotBaseQuals, 24
- plotBaseQuals, ANY-method  
(plotBaseQuals), 24
- plotBaseQuals, character-method  
(plotBaseQuals), 24
- plotBaseQuals, FastqcData-method  
(plotBaseQuals), 24
- plotBaseQuals, FastqcDataList-method  
(plotBaseQuals), 24
- plotDupLevels, 26
- plotDupLevels, ANY-method  
(plotDupLevels), 26
- plotDupLevels, character-method  
(plotDupLevels), 26
- plotDupLevels, FastqcData-method  
(plotDupLevels), 26
- plotDupLevels, FastqcDataList-method  
(plotDupLevels), 26
- plotFastqcPCA, 28
- plotFastqcPCA, ANY-method  
(plotFastqcPCA), 28
- plotFastqcPCA, character-method  
(plotFastqcPCA), 28
- plotFastqcPCA, FastqcDataList-method  
(plotFastqcPCA), 28
- plotGcContent, 30
- plotGcContent, ANY-method  
(plotGcContent), 30
- plotGcContent, character-method  
(plotGcContent), 30
- plotGcContent, FastqcData-method  
(plotGcContent), 30
- plotGcContent, FastqcDataList-method  
(plotGcContent), 30
- plotKmers, 33
- plotKmers, ANY-method (plotKmers), 33
- plotKmers, character-method (plotKmers),  
33
- plotKmers, FastqcData-method  
(plotKmers), 33
- plotKmers, FastqcDataList-method  
(plotKmers), 33
- plotNContent, 35
- plotNContent, ANY-method (plotNContent),  
35
- plotNContent, character-method  
(plotNContent), 35
- plotNContent, FastqcData-method  
(plotNContent), 35
- plotNContent, FastqcDataList-method  
(plotNContent), 35
- plotOverrep, 36
- plotOverrep, ANY-method (plotOverrep), 36
- plotOverrep, character-method  
(plotOverrep), 36
- plotOverrep, FastqcData-method  
(plotOverrep), 36
- plotOverrep, FastqcDataList-method  
(plotOverrep), 36
- plotReadTotals, 38
- plotReadTotals, ANY-method  
(plotReadTotals), 38
- plotReadTotals, character-method  
(plotReadTotals), 38
- plotReadTotals, FastqcDataList-method  
(plotReadTotals), 38
- plotSeqContent, 40
- plotSeqContent, ANY-method  
(plotSeqContent), 40
- plotSeqContent, character-method  
(plotSeqContent), 40
- plotSeqContent, FastqcData-method  
(plotSeqContent), 40
- plotSeqContent, FastqcDataList-method  
(plotSeqContent), 40
- plotSeqLengthDistn, 42
- plotSeqLengthDistn, ANY-method  
(plotSeqLengthDistn), 42
- plotSeqLengthDistn, character-method  
(plotSeqLengthDistn), 42
- plotSeqLengthDistn, FastqcData-method  
(plotSeqLengthDistn), 42
- plotSeqLengthDistn, FastqcDataList-method  
(plotSeqLengthDistn), 42
- plotSeqQuals, 43
- plotSeqQuals, ANY-method (plotSeqQuals),  
43

plotSeqQuals, character-method  
    (plotSeqQuals), [43](#)  
plotSeqQuals, FastqcData-method  
    (plotSeqQuals), [43](#)  
plotSeqQuals, FastqcDataList-method  
    (plotSeqQuals), [43](#)  
plotSummary, [46](#)  
plotSummary, ANY-method (plotSummary), [46](#)  
plotSummary, character-method  
    (plotSummary), [46](#)  
plotSummary, FastqcDataList-method  
    (plotSummary), [46](#)  
pwf, [48](#)  
PwfCols, [21](#), [25](#), [27](#), [32](#), [34](#), [35](#), [37](#), [41](#), [45](#), [47](#)  
PwfCols (PwfCols-class), [48](#)  
PwfCols-class, [48](#)  
  
readTotals, [39](#), [49](#)  
runFastQC, [50](#)  
runFastQC, ANY-method (runFastQC), [50](#)  
runFastQC, BamFile-method (runFastQC), [50](#)  
runFastQC, BamFileList-method  
    (runFastQC), [50](#)  
runFastQC, FastqFile-method (runFastQC),  
    [50](#)  
runFastQC, FastqFileList-method  
    (runFastQC), [50](#)  
  
setAlpha (getColours), [10](#)  
setAlpha, PwfCols-method (getColours), [10](#)  
setColours (getColours), [10](#)  
setColours, PwfCols-method (getColours),  
    [10](#)  
  
TheoreticalGC-class, [53](#)  
  
writeHtmlReport, [53](#)